NAS Technical Report NAS-09-004, November 2009

Large Scale Aerodynamic Calculation on Pleiades

Thomas H. Pulliam, Dennis C. Jespesen

ABSTRACT

A very large scale aerodynamic calculation on the NASA Pleiades supercomputer using the three-dimensional Navier-Stokes code OVERFLOW is presented. The application is a rotorcraft blade simulation, with attention focused on the resolution of the vortices which develop off the blade tips and propagate through the flow domain. This simulation represents a level of refinement beyond any previous results presented for this problem. This extremely large CFD application stresses the capabilities of the Pleiades system, uses a relevant application with high levels of grid resolution to provide benchmark results, and helps to assess the capabilities of the OVERFLOW code in terms of scalability, parallel efficiency and performance.

Keywords: large scale aerodynamics, supercomputer, rotorcraft, OVERFLOW code

1. INTRODUCTION

A very large scale aerodynamic calculation on the NASA Pleiades supercomputer is presented here. There are three main purposes for this effort. First, we define an extremely large CFD application to stress the capabilities of the Pleiades system in terms of job size, file size, memory use, disk traffic, message-passing, and post-processing of the extremely large data sets produced. Second, a relevant application was chosen which at high levels of grid resolution provides benchmark results for comparison with coarser grid results and with new grid adaption techniques. Third, we assess the capabilities of the OVERFLOW code in terms of scalability, parallel efficiency and performance under the stress of an extremely large problem. This work emphasizes the capability of the Pleiades system to perform very large aerodynamic calculations.

The application is a rotorcraft blade simulation, which involves complicated physics and geometry, and is one of the more challenging physical applications in terms of CFD resources. In particular, we focus on the resolution of the

Thomas H. Pulliam, MS T27B-1, NASA/Ames Research Center, Moffett Field, CA 94035, U.S.A., Thomas.H.Pulliam@nasa.gov
Dennis Jespersen, MS 258-2, NASA/Ames Research Center, Moffett Field, CA 94035, U.S.A., Dennis.Jespersen@nasa.gov

vortices which develop off the blade tips and propagate through the flow domain. Insufficient resolution causes the vortices to decay too rapidly and to convect to improper positions. This simulation represents a degree of refinement beyond any previous results presented for this problem.

2. PLEIADES ARCHITECTURE

The Pleiades supercomputer at NASA/Ames Research Center is an SGI Altix ICE system with 5888 nodes and 8 cores per node, for a total of 47104 cores. Each node consists of dual quad-core Intel Xeon EX5472 ("Harpertown") processors. Each quad-core chip has two 3 MB L2 caches, each cache being shared among 2 of the 4 cores on the chip. The cores have a clock frequency of 3.0 GHz with a maximum of 4 floating-point operations per clock per core. The front-side bus frequency is 1600 MHz with a peak transfer rate of 12.8 GB/sec. Most of the nodes have 8 GB of memory (shared among the 8 cores), but there are 65 "bigmem" nodes that each have 16 GB of memory.

The nodes are connected by two InfiniBand fabrics, each fabric supporting both message-passing and I/O traffic. At the time of our tests, all large-scale jobs on the system shared a single Lustre [10] filesystem. Jobs are run in a batch environment with scheduling by the Portable Batch Scheduler (PBS).

3. THE OVERFLOW CODE

Overflow [2, 5] solves the Reynolds-averaged Navier-Stokes (RANS) equations in generalized coordinates on a system of overset grids. The code has options for several solution algorithms; in this work we used third-order central differencing with artificial dissipation terms in space and the diagonalized Beam-Warming implicit three-factor central-difference scheme for time-stepping [6]. The solutions presented here were run in steady-state mode (time accurate capability is available). The code has algebraic, one-equation, and two-equation turbulence models.

The overset grid approach was originated [1] to simplify the spatial discretization of complex geometries using structured grids. A complex geometry is subdivided into a set of components with relatively simple geometry, and an appropriate grid placed around each component. The component grids are overlaid, or overset, to form a composite grid. Trilinear interpolation is used to transfer computed data between grids. Domain connectivity is accomplished using an object X-ray and automatic hole cutting technique, see [3, 4].

Parallelism in Overflow can occur at two levels. At the higher level there is explicit-message passing using MPI. There is an optional lower level of parallelism using OpenMP, which utilizes shared memory. To give good load balance, Overflow will split grids which contain more than some target number of grid points. This target can be user-specified, or if not specified the code chooses the target as half the average number of grid points per MPI rank; in practice this almost always provides good load balance. Each MPI rank is eventually assigned one or more grids (for large-scale problems with good load balance, there are typically 2 to 3 grids per MPI rank). The code then proceeds to march in time, with each time step having two stages. The first stage is communication: each MPI rank sends all the inter-grid data needed by other MPI ranks, and receives all necessary data. The second stage is computation: each MPI rank computes on its grids, with no message-passing involved. After the second stage all MPI ranks send a small amount of information (residuals, minimum pressure and density, etc.) to the MPI rank 0 process. All I/O is via the MPI rank 0 process.

Most OVERFLOW users utilize only the MPI level of parallelism. This is sufficient for problems which use up to a few hundred or so MPI ranks. For very large scale computations with thousands of MPI ranks, using only the outer level of parallelism would result in a tremendous amount of grid splitting and a significant increase in the number of grid points, due to "fringe" (or "halo") points generated during splitting. This increase in grid points would result in increased computation time and increased message-passing during the communication phase of the code; it also would increase memory use. During preliminary work on this project we encountered situations where a given number of cores in pure MPI mode were unable to run a case due to insufficient memory, while the same number of cores in hybrid parallel mode (MPI for outer parallelism among nodes, OpenMP for inner parallelism within a node) could run the case successfully. In the computation presented here we used hybrid mode, with MPI for parallelism at the outer level, among nodes, and OpenMP for parallelism at the inner level within a node.

4. PHYSICAL PROBLEM

The Tilt Rotor Aeroacoustics Model (TRAM), a 1/4-scale three blades and hub component of the V-22 Osprey tiltrotor aircraft, was chosen for this study. Extensive details of the physical problem, geometry, experimental data and previous numerical studies can be found in Potsdam and Strawn [8]. The goal is to compute the steady hover mode of the TRAM blade system. Therefore, the OVERFLOW computations were performed in a steady non-inertial frame where the observer is moving with the blades and the flow appears stationary with respect to the observer. The one-equation Baldwin-Barth turbulence model was used with corrections, Potsdam and Pulliam [7].

The specifics of the grid system which are pertinent to this study are briefly discussed here. The three-bladed TRAM rotor system is shown in Figure 1, along with a slice of the off-body grid system. In terms of the unstructured overset methodology employed by OVERFLOW, near-body (NB) curvilinear structured grids are generated about the blade and hub geometries with sufficent resolution to capture viscous effects. The reference length scale for this problem is the tip chord (C_{tip}) of the blades. Off-body Cartesian grids are automatically generated by OVERFLOW. The "Baseline" case as defined in Potsdam and Strawn uses a Level 1 grid (L_1) which is uniform in all coordinate directions with a grid spacing $\Delta L_1 = 0.1 C_{\text{tip}}$. The near-body grids are completely embedded within the L_1 grid, where the overset methodologies of hole-cutting, iblanking, and Chimera interpolation logic are employed to interface the grid systems [3]. Subsequent off-body Level 2 and higher Cartesian brick grids are generated with spacing $\Delta L_i = 2^{i-1} \Delta L_1$ to extend the grid to the outer boundary of the computational domain, see Figure 1.

The Baseline grid of Potsdam and Strawn is a standard for comparison and will be the basis for our comparisons here. Table I lists the characteristics of

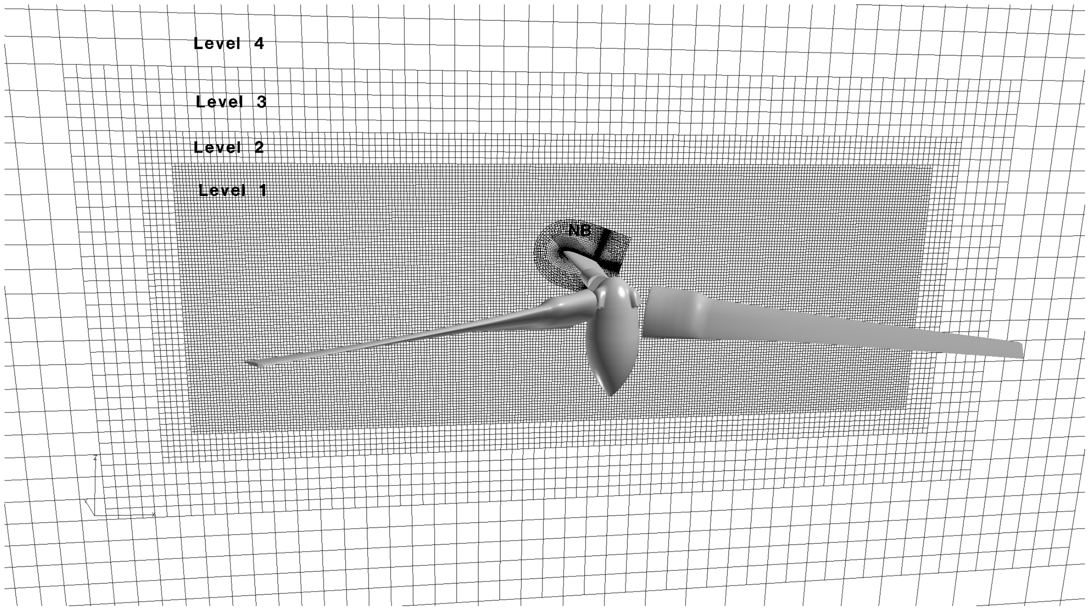

Figure 1. TRAM V22 geometry, Hover

TABLE I. Grid Size for Different Refinement Levels

Case	ΔL_1	No. of grid points	No. of grids
Baseline	$0.1 \cdot C_{\text{tip}}$	13,692,987	50
Medium	$0.05 \cdot C_{\text{tip}}$	58,426,144	70
Large	$0.025 \cdot C_{\text{tip}}$	379,761,734	188
Huge	$0.0125 \cdot C_{\text{tip}}$	3,087,812,624	194

the Baseline grid and a sequence of refined grids. These grids are a sequence of halvings of the Level 1 spacing to improve the resolution of the wake vortices being shed off the blade tips. The "Huge" mesh is the one chosen for the current study which represents three levels of refinement of the Level 1 spacing and produces a grid with over 3 billion points, possibly the largest aerodynamic RANS CFD calculation to date.

5. RESULTS

Shown in Figure 2 is a comparison of the solution on the Baseline grid system with the solution on the refined Huge grid system. The results of the Baseline case are exactly consistent with those reported by Potsdam and Strawn [8] and Potsdam and Pulliam [7]. In Figure 2, iso-surfaces of vorticity show that the refined grid captures the wake vortex more accurately than the Baseline grid, producing significantly less decay of the vortex cores. In the Baseline calculation, at the contour level chosen, the vortices dissipate after turning approximately 90 degrees from the blade tip. In the Huge case, at the same contour level, the vortices persist for more than 360 degrees before dissipating. Also shown in the Huge case is the ability to capture fine details of the shear layers shed from the

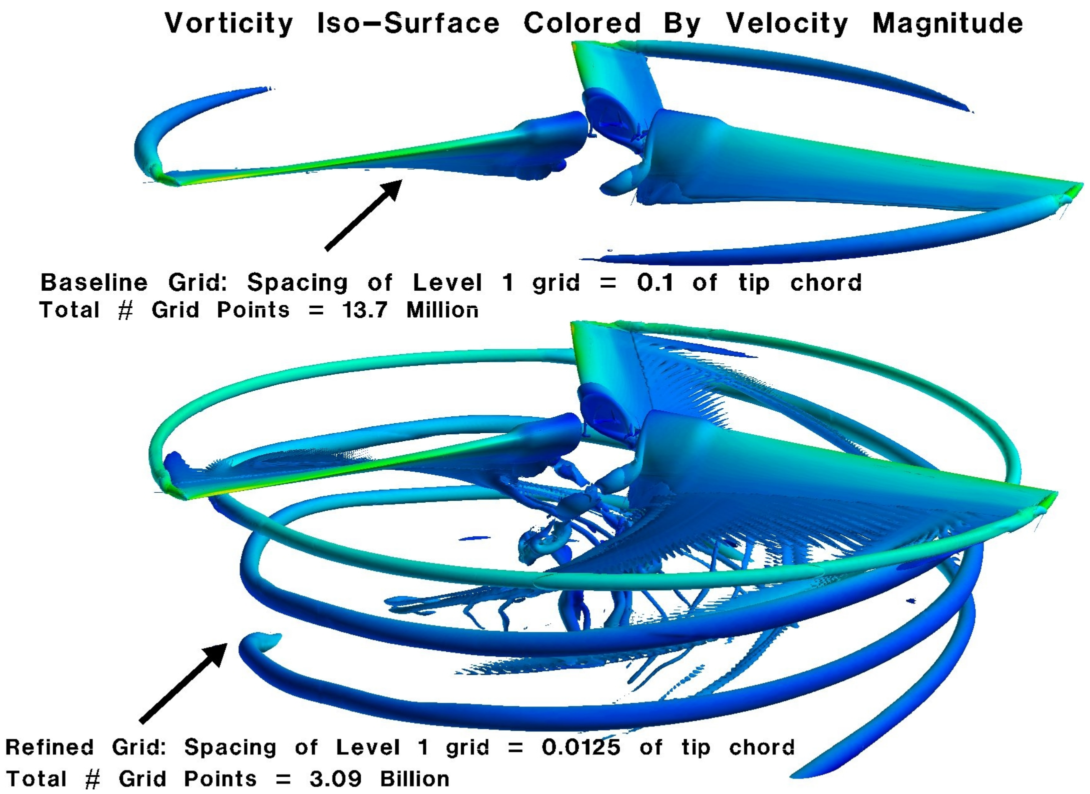
Vorticity Iso–Surface Colored By Velocity Magnitude
Baseline Grid: Spacing of Level 1 grid = 0.1 of tip chord
Total # Grid Points = 13.7 Million
Refined Grid: Spacing of Level 1 grid = 0.0125 of tip chord
Total # Grid Points = 3.09 Billion

Figure 2. TRAM Comparison

blades and the interaction of the shear layers with the tip vortices.

One of the goals of this study is to produce a relevant application at high levels of grid resolution to provide benchmark results for comparison with coarser grid results and with new grid adaption techniques. One measure of the effect of increased resolution on the capturing of rotor wake vortices is the growth rate of the vortex cores as a function of azimuthal angle measured from one of the blade tips. Define the vortex core diameter as the distance between the minimum and maximum cross-flow velocity components (normalized by blade tip chord). Then the growth of the vortex core as a function of wake age can be constructed using the velocity profiles and is displayed in Figure 3. Also shown is a composite of experimental data, from Holst and Pulliam [9]. The results of a grid refinement sequence demonstrate marked improvement of the rate of decay of the vortex core. This culminates in the Huge grid giving a rate comparable to the experimental data.

6. OBSERVATIONS

Table II shows some performance data for Overflow on the Huge case. In the table under Layout, "$m \times n$" means m MPI ranks with each MPI rank having n OpenMP threads. The grid has 194 zones with 3,087,812,624 total points.

Performance is in terms of nanoseconds per grid point per timestep, so lower is better. This metric is computed both in terms of the nominal number of grid

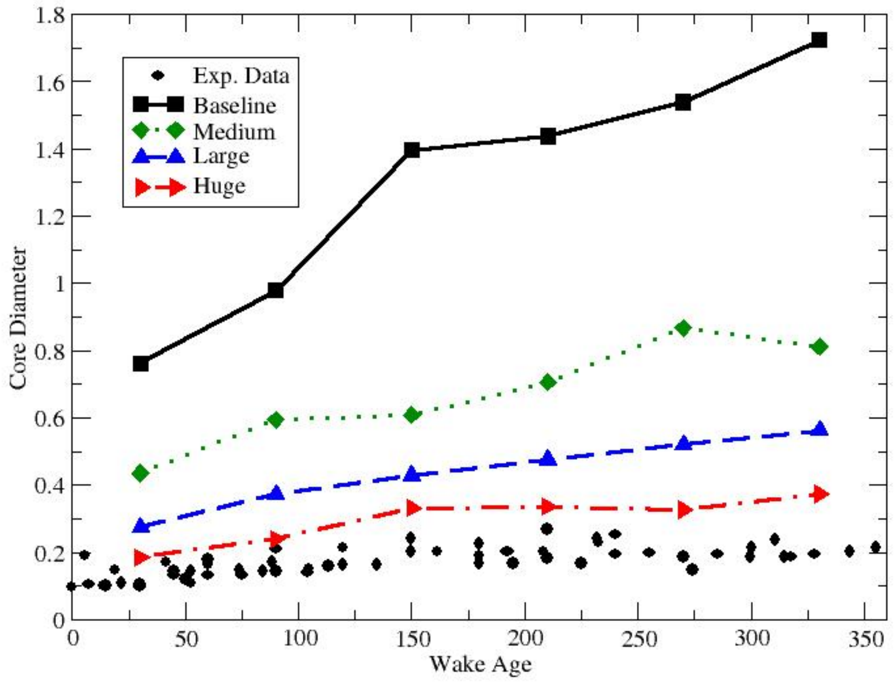

Figure 3. Growth Rate of Vortex Core with Wake Age

points (3,087,812,624) and the actual number of grid points after splitting. The 128×8 case did not run due to insufficient memory. All these runs were made in a normal shared environment (not in dedicated time), and all were made with the MPI rank 0 process on a "bigmem" node. Notice the increase in number of zones and number of grid points with the increasing number of MPI ranks.

The last entry in Table 2 represents 16384 cores, with the original 194 zones increasing to over 6000 zones from splitting, and a 15% increase in the number of points. Without the use of OpenMP threads, 16384 MPI ranks would produce over 48,000 zones and a 36% increase in the total number of points. The negative impact of this increased number of grid points on performance, and the increased memory footprint would be significant, so it is advantageous for us to use the

TABLE II. Overflow Performance for Huge Case

	after splitting		nanosec/pt/timestep	
Layout	no. zones	no. pts	nominal pts	actual pts
128×8	322	3129428624	N/A	N/A
192×8	580	3189903794	8.815	8.533
256×8	744	3209504544	7.047	6.780
512×8	1565	3290837114	2.951	2.769
1024×8	3227	3405635864	1.242	1.126
2048×8	6880	3550830749	0.574	0.499

hybrid (MPI+OpenMP) code.

Notice also the good scaling of the code as the number of MPI ranks is increased while the number of OpenMP threads is fixed. The observed superlinear scaling is probably due to cache effects, with more total cache memory available at higher node counts. Message-passing takes a small portion of total time for all these cases.

The result shown in section 5 was computed with the 2048×8 processor layout and was run for 40000 time steps. Walltime per step was about 1.77 seconds; at this computational rate 40000 steps took about 20 wallclock hours (spread over several runs). This does not include startup, shutdown, and checkpoint time which added about another 4 wallclock hours. Time to write a solution (including sending all data to MPI rank 0) was in the 16-17 minute range.

Some early attempts to run cases with 16384 cores ran into various problems. A common problem was difficulty in the software underlying the MPI library, which was either nonoperational or appeared to be nonoperational, causing the job to fail to start. A particular issue with large node-count jobs is the increased possibility of single-point hardware or software problems, causing the whole job to fail.

The solution file for the Huge case is 173 GB in size and the grid file is 86 GB. Disk space restrictions made it important to quickly copy data off Pleiades, but at the time these computations were performed the maximum data transfer rate to archival storage was about 115 MB/sec, and often rates an order of magnitude less were seen. This rate made expeditious transfer of large files tedious. (Since this work was performed, various system improvements have significantly increased the file transfer rate to archival storage to about 300 MB/sec.)

This work illustrates the capability of the Pleiades system to handle very large aerodynamic calculations. The solution was obtained in less than 24 hours of total wallclock time, which for a problem of this size is a significant achievement.

REFERENCES

1. Benek, J. A., Buning, P. G., and Steger, J. L. 1985. “A 3-D Chimera Grid Embedding Technique,” *AIAA 7th Computational Fluid Dynamics Conference*, Cincinnati, OH, paper no. AIAA-85-1523.
2. Buning, P. G., Chiu. I. T., Obayashi, S., Rizk, Y. M., and Steger, J. L. 1988. “Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent,” *AIAA Atmospheric Flight Mechanics Conference*, Minneapolis, MN, paper no. AIAA-1988-4359.
3. Chan, W. M. 2009. “Overset Grid Technology Development at NASA Ames Research Center.” *Computers and Fluids*, 38(3):496-503.
4. Meakin, R. L. 2001. “Automatic Off-body Grid Generation for Domains of Arbitrary Size,” *AIAA 15th Computational Fluid Dynamics Conference*, Anaheim, CA, paper no. AIAA-2001-2536.
5. Nichols, R. H., Tramel, R. W., and Buning, P. G. 2006. “Solver and Turbulence Model Upgrades to OVERFLOW 2 for Unsteady and High-speed Applications,” *AIAA 24th Applied Aerodynamics Conference*, San Francisco, CA, paper no. AIAA-2006-2824.
6. Pulliam, T. H. and Chaussee, D. S. 1981. “A Diagonalized Form of an Implicit Approximate Factorization Algorithm,” *J. Comp. Phys.*, 39(2):347-363.
7. Potsdam, M. A. and Pulliam, T. H. 2008. “Turbulence Modeling Treatment for Rotocraft Wakes,” presented at the AHS Aeromechanics Specialist’s Meeting, San Francisco, CA, January 23-25, 2008.
8. Potsdam, M. A. and Strawn, R. C. 2005. “CFD Simulations of Tiltrotor Configurations in

Hover," *Journal of the American Helicopter Society*, 50(1):82-94.

9. Holst, T. L. and Pulliam, T. H. 2009. "Overset Solution Adaptive Grid Approach Applied to Hovering Rotorcraft Flows," *AIAA 27th Applied Aerodynamics Conference, San Antonio, Texas*, Paper No. AIAA-2009-3519.
10. `http://www.lustre.org.`